UNITED STATES PATENT AND TRADEMARK OFFICE

mn

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/727,183	12/03/2003	Miroslav Cina	13913-127001 / 2003P00384	5201

32864 7590 04/09/2007
FISH & RICHARDSON, P.C.
PO BOX 1022
MINNEAPOLIS, MN 55440-1022

EXAMINER	
TRUONG, CAM Y T	
ART UNIT	**PAPER NUMBER**
2162	

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	04/09/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

Office Action Summary	**Application No.** 10/727,183	**Applicant(s)** CINA, MIROSLAV	
	Examiner Cam Y T. Truong	**Art Unit** 2162	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *28 September 2006*.

2a)☒ This action is **FINAL**. 2b)□ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-21 and 26-40* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-21, 26-40* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____.

5) □ Notice of Informal Patent Application

6) □ Other: _____.

DETAILED ACTION

1. Applicant has amended claims 1, 9, 14, 26, 31, 34, canceled claims 22-25 and added claims 39-40 in the amendment filed on 9/28/2006.

Claims 1-21 and 26-40 are pending in this office action.

Response to Arguments

2. Applicant's arguments with respect to claims 1-21, 26-40 have been considered but are moot in view of the new ground(s) of rejection.

Applicant argued that claim 1 is clear and definite and that word "may" in claim 1 does not cause the claim to be indefinite.

In response to applicant's argument, the word "may" is unclear whether the previous lock level can be associated with other processes or cannot be associated with other processes. Thus, word "may" is being indefinite for failing to particularly point out distinctly claim the subject matter which applicant regards as the invention.

Applicant argued Brenner does not teach the claimed limitation "the particular process repeatedly attempting to associate the particular process with a lower lock level; repeatedly attempting to associating itself with a lower lock level".

In response to applicant's argument, claims 1-21, 26-40 have been considered but are moot in view of the new ground(s) of rejection. Applicant's election with traverse of group I in the reply filed on 1/12/2007 is acknowledged. The traversal is on the ground(s) . This is found persuasive. Thus, the restriction is withdraw.

Claim Rejections - 35 USC § 112

3. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

4. Claims 1 and 31 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. The word "may" in claim 1, line 8; in claim 31, line 9 the word "may" is unclear whether the previous lock level can be associated with other processes or cannot be associated with other processes. Thus, word "may" is being indefinite for failing to particularly point out distinctly claim the subject matter which applicant regards as the invention.

Claims 2-8, 32-33 are dependent on claims 1 and 31; thus, they are rejected under the same rational.

Claim Rejections - 35 USC § 101

5. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

6. Claims 1-8, 14-18, 39 are rejected under 35 U.S.C.101 because the language of the claim raises a question as to whether the claim is directed merely to an abstract

idea that is not tied to a technological art, environment or machine which would result in a practice application producing a concrete, useful, and tangible result to form the basis of statutory subject matter under 35 U.S.C 101.

Claims 1-8, 14-18, 39 recite "a method". The claims are rejected as falling under the judicial exception of an abstract idea which lacks a useful, concrete, and tangible result. A claimed series of steps or acts that do not result in a useful, concrete, and tangible result are not statutory within the meaning of 35 USC 101. In the instant case, the claim 1 recite, "_[associating]_," "_[attempting]_," "_[allowing]_." The claim 14 recite "locking", "assigning" and "permitting". However, no useful, concrete, and tangible result is claimed. For example, "writing said data," "updating said data," "sending said data" being claimed at the end of the claim may comprise a useful, concrete, and tangible result. Absent such a result, however, the claims are not statutory.

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

8. Claims 1-21 and 26-40 are rejected under 35 U.S.C. 103(a) as being unpatentable over Brenner et al (or hereinafter "Brenner") (US 2002/0078119) in view of Frank (US 5790851).

As to claim 1, Brenner teaches the claimed limitations:

In a software –implemented procedure associating a lock level with a particular process, a higher lock level representing a larger number of other processes having priority over the particular process in accessing the table (paragraphs [0020, 0050]);

attempting to associate the particular process with a lower lock level, and if the particular process has been successfully associated with the lower lock level, releasing a previous lock level associated with the particular process so that the previous lock level may be associated with other processes (paragraph [0053]); and

allowing the particular process to access the table when the lock level for the particular process is equal to a preset value (paragraphs [0053, 0048]).

Brenner does not explicitly teach the claimed limitation " each of the processes being associated with no more than one lock level; the particular process repeatedly". Frank teaches each processes being associated with one lock level and the process repeated to associate with lock levels (figs. 3-5, col. 7, lines 5-67; col. 8, lines 15-30).

It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Frank's teaching of each processes being associated with one lock level and the process repeated to associate with lock levels to Brenner's system in order to minimizes the number of context switches required to process lock requests, by providing direct lock access to user processes.

As to claims 2 and 32, Brenner teaches the claimed limitations in which the preset value is equal to one (paragraphs [0053, 0048]).

As to claims 3 and 33, Brenner teaches the claimed limitations “in which each of the processes attempts to associate itself with a lower lock level independently of other processes” as (paragraphs [0050; 0054]).

As to claim 4, Brenner teaches the claimed limitations “further comprising storing in a queue information indicating which process is associated with which lock level” as (paragraphs [0050-0052]).

As to claim 5, Brenner teaches the claimed limitations “calling multiple instances of a procedure that associates a lock level with a process, each instance of the procedure associated with one of the multiple processes and is configured to attempt to associate a different lock level with the process associated with the instance until the process is granted access to the record” as (paragraphs [0050-0052], fig. 1).

As to claim 6, Brenner teaches the claimed limitations “allowing processes to read the record but not modify the record when the lock levels for the processes are different from the preset value” as (paragraph 0054, fig. 3).

As to claim 7, Brenner teaches the claimed limitations “locking the record when the lock level having the preset value is associated with a process” as (paragraphs [0048, 0053], fig. 3).

As to claim 8, Brenner teaches the claimed limitations “in which at least two of the processes are being run in a parallel processing environment” as (paragraph [0018]).

As to claim 9, Brenner teaches the claimed limitations:

In a software-implemented procedure, upon receiving a request from a first process to access a record in a database, associating a first lock level with the first process and allowing the first process to access the record, preventing other processes from modifying the record until the first process finishes accessing the record (paragraph [0059; 0018]);

upon receiving a request from a second process to access the record while the first process is still accessing the record, associating a second lock level with the second process (paragraphs [0053-0050]);

upon receiving a request from a third process to access the record while the first process is still access the record, associating a third lock level with the third process, when the first process finishes accessing the record, releasing the first lock level, when one of the second and third processes associates itself with the first lock level, permitting the process to modify the record (paragraphs 0053, 0050, 0018)

Brenner does not explicitly teach the claimed limitation “the second and third processes each repeatedly attempting to associate itself with a lower lock level ”.

Frank teaches each processes being associated with one lock level and the process repeated to associate with lock levels (figs. 3-5, col. 7, lines 5-67; col. 8, lines 15-30).

It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Frank's teaching of each processes being associated with one lock level and the process repeated to associate with lock levels to Brenner's system in order to minimizes the number of context switches required to process lock requests, by providing direct lock access to user processes.

As to claims 10 and 27, Brenner teaches the claimed limitation "in which preventing other processes from modifying the record comprises allowing the other processes to read the record but not modify the record" as (fig. 3, paragraph [0054]).

As to claims 11 and 28, Brenner teaches the claimed limitation "locking the record when the first lock level is associated with a process" as (fig. 3, paragraph [0054]).

As to claims 12 and 29, Brenner teaches the claimed limitation "writing to a queue to specify which lock level is associated with which process" as (paragraph [0054], fig. 3).

As to claims 13 and 30, Brenner teaches the claimed limitation "in which at least two of the first, second, and third processes are being run in a parallel processing environment" as (paragraph [0018]).

As to claims 14 and 34, Brenner teaches the claimed limitation:

In a software implemented procedure locking a record in a database at multiple levels when multiple processes running in parallel attempt to access the record (paragraphs [0018, 0048]);

assigning a lock level to each of the multiple processes, each process having a different lock level (paragraph [0053]); and

selectively permitting one of the multiple processes to access the record at a time (paragraph [0054]).

Brenner does not explicitly teach "each of the process not currently associated with a lock level repeatedly attempting to associated itself with a lower lock level". Frank teaches each processes being associated with one lock level and the process repeated to associate with lock levels (figs. 3-5, col. 7, lines 5-67; col. 8, lines 15-30).

It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Frank's teaching of each processes being associated with one lock level and the process repeated to associate with lock levels to Brenner's system in order to minimizes the number of context switches required to process lock requests, by providing direct lock access to user processes.

As to claims 15 and 35, Brenner teaches the claimed limitation "reassigning the lock levels of the processes when a process accessing the record terminates its access to the record" as (paragraphs [0069-0070]).

As to claims 16 and 36, Brenner teaches the claimed limitation "in which a process that attempted to access the record earlier than another process is assigned a lower lock level than the other process, and each process other than the process terminating its access to the record is assigned a lower lock level when the process terminates its access to the record" as (paragraphs [0053, 0054]).

As to claims 17 and 37, Brenner teaches the claimed limitation "storing in a queue information indicating which process is associated with which lock level" as (fig. 2).

As to claims 18 and 38, Brenner teaches the claimed limitation "calling multiple instances of a procedure that assigns a lock level to a process, each instance of the procedure associated with one of the multiple processes and is configured to attempt to assign a different lock level to the process until the process is granted access to the record" as (paragraphs [0050-0052]).

As to claim 19, Brenner teaches the claimed limitations:

A database to store records (paragraph [0018]); and

a queue to store information relating to lock levels of processes that attempt to access the records, each different process having a different lock level when attempting to access the same record, one of the processes having a particular lock level, the process having a particular lock level being allowed to access the record (paragraphs [0053, 0050]);

the procedure allowing any one process to access record when that one process has the particular lock level (paragraph 0050-0052).

Brenner does not explicitly teach the claimed limitation:

"a programmable processor to execute a procedure to assign a respective lock level to each of different processes, each of the different processes having a lock level other than the particular lock level repeatedly attempting to associated itself with another lock level that closer to the particular lock level ".

Frank teaches each processes being associated with one lock level and the process repeated to associate with lock levels (figs. 3-5, col. 7, lines 5-67; col. 8, lines 15-30).

It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Frank's teaching of each processes being associated with one lock level and the process repeated to associate with lock levels to Brenner's system in order to minimizes the number of context switches required to process lock requests, by providing direct lock access to user processes.

As to claim 20, Brenner teaches the claimed limitation "a memory to store software code for implementing a procedure in which instances of the procedure are used to assign lock levels to the processes" as (paragraphs [0050-0052]).

As to claim 21, Brenner teaches the claimed limitation "in which the software code is configured so that the instances of the procedure are run in parallel" as (paragraph [0018]).

As to claim 26, Brenner teaches the claimed limitations:

Upon receiving a request from a first process to access a record a database, associate a first lock level with the first process and allow the first process to access the record but prevent other processes from accessing the record until the first process finishes accessing the record (paragraphs [0018, 0048]);

Upon receiving a request from a second process to access record while the first process still accessing the record associate a second lock level with the second process upon receiving a request from a third process to access the record while the first process still accessing the record, associating a third lock level with the third process (paragraphs 0053-0054);

When the first process finishes accessing the record, release the first lock level from being associated with first process (paragraph 0053);

Brenner does not explicitly teach the claimed limitation "cause of each of second and third process independently attempt to associated itself with a lower lock level and when one of second and third processes associated itself with the first lock level, permitting the process access the record".

Frank teaches each processes being associated with one lock level and the process repeated to associate with lock levels (figs. 3-5, col. 7, lines 5-67; col. 8, lines 15-30).

It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Frank's teaching of each processes being associated with one lock level and the process repeated to associate with lock levels to Brenner's system in order to minimizes the number of context switches required to process lock requests, by providing direct lock access to user processes.

Claim 31 is rejected under the same reason as discussed in claim 1.

Claim 34 is rejected under the same reason as discussed in claim 14.

As to claims 39 and 40, Brenner teaches the claimed limitation "different processes operate in parallel to attempt associate with lower lock levels" as (paragraph 0053).

9. Claims 1-21 and 26-40 are rejected under 35 U.S.C. 103(a) as being unpatentable over Brenner et al (or hereinafter "Brenner") (US 2002/0078119) in view of Mckenney (US 6823511).

As to claim 1, Brenner teaches the claimed limitations:

In a software –implemented procedure associating a lock level with a particular process, a higher lock level representing a larger number of other processes having priority over the particular process in accessing the table (paragraphs [0020, 0050]);

attempting to associate the particular process with a lower lock level, and if the particular process has been successfully associated with the lower lock level, releasing a previous lock level associated with the particular process so that the previous lock level may be associated with other processes (paragraph [0053]); and

allowing the particular process to access the table when the lock level for the particular process is equal to a preset value (paragraphs [0053, 0048]).

Brenner does not explicitly teach the claimed limitation " each of the processes being associated with no more than one lock level; the particular process repeatedly".

Mckenney a mutual exclusion lock allows only the processor holding the lock to execute an associated action. When a processor requests a mutual exclusion lock, it is granted to that processor exclusively. Other processors desiring the lock must wait until the processor with the lock releases it. To solve the add-one-to-a-counter scenario described above, for example, both the first and the second processors would request the mutual exclusion lock before executing further. Whichever processor first acquires the lock then reads the counter, increments the register, and writes to the counter before releasing the lock. The other processor must wait until the first processor finishes and releases the lock; it then acquires the lock, performs its operations on the counter, and releases the lock. In this way, the lock guarantees the counter is incremented twice if the instructions are run twice, even if processors running in parallel execute the (col. 2, lines 35-55).

It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Mckenney's teaching of a processor repeatedly associating with a lock to Brenner's system in order to control the order of process execution for performing any updating without other applications potentially attempting to update the same portion of the database at the same time and minimizes the number of context switches required to process lock requests, by providing direct lock access to user processes.

As to claims 2 and 32, Brenner teaches the claimed limitations in which the preset value is equal to one (paragraphs [0053, 0048]).

As to claims 3 and 33, Brenner teaches the claimed limitations "in which each of the processes attempts to associate itself with a lower lock level independently of other processes" as (paragraphs [0050; 0054]).

As to claim 4, Brenner teaches the claimed limitations "further comprising storing in a queue information indicating which process is associated with which lock level" as (paragraphs [0050-0052]).

As to claim 5, Brenner teaches the claimed limitations "calling multiple instances of a procedure that associates a lock level with a process, each instance of the procedure associated with one of the multiple processes and is configured to attempt to associate a different lock level with the process associated with the instance until the process is granted access to the record" as (paragraphs [0050-0052], fig. 1).

As to claim 6, Brenner teaches the claimed limitations "allowing processes to read the record but not modify the record when the lock levels for the processes are different from the preset value" as (paragraph 0054, fig. 3).

As to claim 7, Brenner teaches the claimed limitations "locking the record when the lock level having the preset value is associated with a process" as (paragraphs [0048, 0053], fig. 3).

As to claim 8, Brenner teaches the claimed limitations "in which at least two of the processes are being run in a parallel processing environment" as (paragraph [0018]).

As to claim 9, Brenner teaches the claimed limitations:

In a software-implemented procedure, upon receiving a request from a first process to access a record in a database, associating a first lock level with the first process and allowing the first process to access the record, preventing other processes from modifying the record until the first process finishes accessing the record (paragraph [0059; 0018]);

upon receiving a request from a second process to access the record while the first process is still accessing the record, associating a second lock level with the second process (paragraphs [0053-0050]);

upon receiving a request from a third process to access the record while the first process is still access the record, associating a third lock level with the third process, when the first process finishes accessing the record, releasing the first lock level, when one of the second and third processes associates itself with the first lock level, permitting the process to modify the record (paragraphs 0053, 0050, 0018).

Brenner does not explicitly teach the claimed limitation "the second and third processes each repeatedly attempting to associate itself with a lower lock level ".

Mckenney a mutual exclusion lock allows only the processor holding the lock to execute an associated action. When a processor requests a mutual exclusion lock, it is granted to that processor exclusively. Other processors desiring the lock must wait until the processor with the lock releases it. To solve the add-one-to-a-counter scenario described above, for example, both the first and the second processors would request the mutual exclusion lock before executing further. Whichever processor first acquires the lock then reads the counter, increments the register, and writes to the counter before releasing the lock. The other processor must wait until the first processor finishes and releases the lock; it then acquires the lock, performs its operations on the counter, and releases the lock. In this way, the lock guarantees the counter is incremented twice if the instructions are run twice, even if processors running in parallel execute the (col. 2, lines 35-55).

It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Mckenney's teaching of a processor repeatedly associating with a lock to Brenner's system in order to control the order of process execution for performing any updating without other applications potentially attempting to update the same portion of the database at the same time and minimizes the number of context switches required to process lock requests, by providing direct lock access to user processes.

As to claims 10 and 27, Brenner teaches the claimed limitation "in which preventing other processes from modifying the record comprises allowing the other processes to read the record but not modify the record" as (fig. 3, paragraph [0054]).

As to claims 11 and 28, Brenner teaches the claimed limitation "locking the record when the first lock level is associated with a process" as (fig. 3, paragraph [0054]).

As to claims 12 and 29, Brenner teaches the claimed limitation "writing to a queue to specify which lock level is associated with which process" as (paragraph [0054], fig. 3).

As to claims 13 and 30, Brenner teaches the claimed limitation "in which at least two of the first, second, and third processes are being run in a parallel processing environment" as (paragraph [0018]).

As to claims 14 and 34, Brenner teaches the claimed limitation:

In a software implemented procedure locking a record in a database at multiple levels when multiple processes running in parallel attempt to access the record (paragraphs [0018, 0048]);

assigning a lock level to each of the multiple processes, each process having a different lock level (paragraph [0053]); and

selectively permitting one of the multiple processes to access the record at a time (paragraph [0054]).

Brenner does not explicitly teach "each of the process not currently associated with a lock level repeatedly attempting to associated itself with a lower lock level".

Mckenney a mutual exclusion lock allows only the processor holding the lock to execute an associated action. When a processor requests a mutual exclusion lock, it is granted to that processor exclusively. Other processors desiring the lock must wait until the processor with the lock releases it. To solve the add-one-to-a-counter scenario described above, for example, both the first and the second processors would request the mutual exclusion lock before executing further. Whichever processor first acquires the lock then reads the counter, increments the register, and writes to the counter before releasing the lock. The other processor must wait until the first processor finishes and releases the lock; it then acquires the lock, performs its operations on the counter, and releases the lock. In this way, the lock guarantees the counter is incremented twice if the instructions are run twice, even if processors running in parallel execute the (col. 2, lines 35-55). It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Mckenney's teaching of a processor repeatedly associating with a lock to Brenner's system in order to control the order of process execution for performing any updating without other applications potentially attempting to update the same portion of the database at the same time and minimizes the number of context switches required to process lock requests, by providing direct lock access to user processes.

As to claims 15 and 35, Brenner teaches the claimed limitation "reassigning the lock levels of the processes when a process accessing the record terminates its access to the record" as (paragraphs [0069-0070]).

As to claims 16 and 36, Brenner teaches the claimed limitation "in which a process that attempted to access the record earlier than another process is assigned a lower lock level than the other process, and each process other than the process terminating its access to the record is assigned a lower lock level when the process terminates its access to the record" as (paragraphs [0053, 0054]).

As to claims 17 and 37, Brenner teaches the claimed limitation "storing in a queue information indicating which process is associated with which lock level" as (fig. 2).

As to claims 18 and 38, Brenner teaches the claimed limitation "calling multiple instances of a procedure that assigns a lock level to a process, each instance of the procedure associated with one of the multiple processes and is configured to attempt to assign a different lock level to the process until the process is granted access to the record" as (paragraphs [0050-0052]).

As to claim 19, Brenner teaches the claimed limitations:

A database to store records (paragraph [0018]); and

a queue to store information relating to lock levels of processes that attempt to access the records, each different process having a different lock level when attempting to access the same record, one of the processes having a particular lock level, the process having a particular lock level being allowed to access the record (paragraphs [0053, 0050]);

the procedure allowing any one process to access record when that one process has the particular lock level (paragraph 0050-0052).

Brenner does not explicitly teach the claimed limitation:

"a programmable processor to execute a procedure to assign a respective lock level to each of different processes, each of the different processes having a lock level other than the particular lock level repeatedly attempting to associated itself with another lock level that closer to the particular lock level ".

Mckenney a mutual exclusion lock allows only the processor holding the lock to execute an associated action. When a processor requests a mutual exclusion lock, it is granted to that processor exclusively. Other processors desiring the lock must wait until the processor with the lock releases it. To solve the add-one-to-a-counter scenario described above, for example, both the first and the second processors would request the mutual exclusion lock before executing further. Whichever processor first acquires the lock then reads the counter, increments the register, and writes to the counter before releasing the lock. The other processor must wait until the first processor finishes and releases the lock; it then acquires the lock, performs its operations on the counter, and releases the lock. In this way, the lock guarantees the counter is incremented twice if the instructions are run twice, even if processors running in parallel execute the (col. 2, lines 35-55).
It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Mckenney's teaching of a processor repeatedly associating with a lock to Brenner's system in order to control the order of process execution for performing any updating without other applications potentially attempting to update the same portion of the database at the same time and minimizes the

number of context switches required to process lock requests, by providing direct lock access to user processes.

As to claim 20, Brenner teaches the claimed limitation "a memory to store software code for implementing a procedure in which instances of the procedure are used to assign lock levels to the processes" as (paragraphs [0050-0052]).

As to claim 21, Brenner teaches the claimed limitation "in which the software code is configured so that the instances of the procedure are run in parallel" as (paragraph [0018]).

As to claim 26, Brenner teaches the claimed limitations:

Upon receiving a request from a first process to access a record a database, associate a first lock level with the first process and allow the first process to access the record but prevent other processes from accessing the record until the first process finishes accessing the record (paragraphs [0018, 0048]);

Upon receiving a request from a second process to access record while the first process still accessing the record associate a second lock level with the second process upon receiving a request from a third process to access the record while the first process still accessing the record, associating a third lock level with the third process (paragraphs 0053-0054);

When the first process finishes accessing the record, release the first lock level from being associated with first process (paragraph 0053);

Brenner does not explicitly teach the claimed limitation "cause of each of second and third process independently attempt to associated itself with a lower lock level and when one of second and third processes associated itself with the first lock level, permitting the process access the record".

Mckenney a mutual exclusion lock allows only the processor holding the lock to execute an associated action. When a processor requests a mutual exclusion lock, it is granted to that processor exclusively. Other processors desiring the lock must wait until the processor with the lock releases it. To solve the add-one-to-a-counter scenario described above, for example, both the first and the second processors would request the mutual exclusion lock before executing further. Whichever processor first acquires the lock then reads the counter, increments the register, and writes to the counter before releasing the lock. The other processor must wait until the first processor finishes and releases the lock; it then acquires the lock, performs its operations on the counter, and releases the lock. In this way, the lock guarantees the counter is incremented twice if the instructions are run twice, even if processors running in parallel execute the (col. 2, lines 35-55).

It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Mckenney's teaching of a processor repeatedly associating with a lock to Brenner's system in order to control the order of process execution for performing any updating without other applications potentially attempting to update the same portion of the database at the same time and minimizes the number of context switches required to process lock requests, by providing direct lock access to user processes.

Claim 31 is rejected under the same reason as discussed in claim 1.

Claim 34 is rejected under the same reason as discussed in claim 14.

As to claims 39 and 40, Brenner teaches the claimed limitation "different processes operate in parallel to attempt associate with lower lock levels" as (paragraph 0053).

Conclusion

10. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL.** See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Contact Information

11. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Cam Y T. Truong whose telephone number is (571) 272-4042. The examiner can normally be reached on Monday to Firday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Breene can be reached on (571) 272-4107. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Cam Y Truong
Primary Examiner
Art Unit 2162
3/30/2007